UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/809,147	03/25/2004	Anson Horton	MS302712.1/MSFTP579US	7392

27195 7590 04/29/2009

TUROCY & WATSON, LLP
127 Public Square
57th Floor, Key Tower
CLEVELAND, OH 44114

EXAMINER	
LEE, MARINA	

ART UNIT	PAPER NUMBER
2192	

NOTIFICATION DATE	DELIVERY MODE
04/29/2009	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

docket1@thepatentattorneys.com
hholmes@thepatentattorneys.com
lpasterchek@thepatentattorneys.com

Office Action Summary	Application No.	Applicant(s)	
	10/809,147	HORTON ET AL.	
	Examiner	Art Unit	
	MARINA LEE	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *10 February 2009*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-6,8-23,26 and 27* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-6,8-23,26, and 27* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *02/27/2009*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Applicants' amendment and response dated February 10, 2009 in responding to the Office Action of December 02, 2008 provided in the rejection of all pending claims 1-6, 8-23, 26, and 27.

Claims 1, 21, 26, and 27 have been amended.

No claims have been canceled nor newly added.

Thus, claims 1-6, 8-23, 26, and 27 are presented for the examination.

2. Applicants' amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP §706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Prior Art's Arguments - Rejections

3. Applicant's arguments with respect to new claim limitation "...examines a display proxy in place of object..." currently recites per independent claims 1, 21,

26, and 27 respectively – *See Remarks, pages 8-10*, have been fully considered but are moot in view of Kobayashi (US 6, 633,888 B1 made of record) as will further be addressed under the Claim Rejections as set forth below.

Claim Rejections - 35 USC § 101

4. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

5. Claims 1-6, 8-20, 26, and 27 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

As per independent claims 1, 26, and 27 recites "A ...debugging system comprising: a debugger….an expression evaluator…" does not comprise a readable medium or any computer hardware (no physical transformation) in order to realize the functionality of the system. The "system" (*i.e., as used in this application, the terms… "system" ….are intended to refer to either … software, or software execution – see Application' specification page 5: lines 3-5)* without such a computer hardware and /or computer readable storage medium may be broadly interpreted as data structures representing descriptive material per ser or computer programming representing computer listing per se – functional descriptive material under 35 USC § 101. *See MPEP 2106.01(I).*

Claims 2-6 and 8-20 recite the limitations that do not cure the deficiency of the base claim 1, which regarding to the rejection of non-statutory under 35 USC 101. Therefore, they are also rejected for the same reasons.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 1-6, 8-18, 21-23, 26, and 27 are rejected under 35U.S.C. 103(a) as being unpatentable over Dandoy, (U.S. Patent No. 2004/0230954 A1 of record) in view of Kobayashi (US 6, 633,888 B1 made of record).

As to claim 1, Dandoy discloses a computer-implemented attributed debugging system (*e.g., system of Fig. 1 – see at least [0017]*) comprising:

a debugger (*e.g., UI Debugger such as 110 or 200 of Figs 1 and 2)* that facilitates debugging of a computer software application (e.g.., application 100 (fig.1)), the debugger obtain values of one or more properties of an object of the computer software application--*(e.g., the debug agent 115 is configured to collect execution data (i.e. object properties…) relating to the graphical user interface 105 during run-time, where the execution data can be obtained via reading state values stored in memory that are maintained by the application during execution – See at least [0017], [0018], and [0029] with emphasis added*);

a variable display component that presents to a developer, values associated with the subset of the one or more properties of the object exposed by the display proxy –(*e.g., object identity such name/type or hierarchy of objects can be determined and displayed upon user's requests – see at least [0046]*).

It is noted that Dandoy does not explicitly disclose an expression evaluator, associated with the debugger, that examines a display proxy in place of the object, the display proxy is implemented as a private nested class of the object, the display proxy is configured to expose a subset of the one or more properties of the object, the subset excludes implementation-specific properties of the object; however, Kobayashi, in analogous art, teaches

"In accordance with one illustrative embodiment, a proxy component (object proxy) is created for each method, including constructors (nested classes), in the component class code, which proxy component encapsulates the parameters of that method. In particular, parameters associated with a method are represented by properties of the proxy component created for that method. When each proxy component is displayed on the GUI of a conventional visual builder, its properties, and consequently, the parameters of the underlying method, are visually editable and can be bound visually to other component properties using, for example, a mechanism in a conventional visual builder which links objects. Exceptions which occur during operation of the method are treated as events and can be visually passed to other components. Therefore, a program can be visually constructed and parameterized for runtime operation." – *See Kobayashi at least col. 5: 1-30, col. 7: 61-67, col. 8: 1-32, col. 4: 43-67 with emphasis added.*

Thus, it would have been obvious to one ordinary skill in the art to use visually creating and testing object of Kobayashi in Debug Agent 115 of Dandoy by creating proxy object and edit/test/ modifying object properties in user interface 105 for increasing programmer efficiency and reduce the time it takes to develop code as taught in Kobayashi (e.g. at least col. 4: 43-60).

As to claim 2, modified Dandoy with Kobayashi discloses the expression evaluator evaluates an expression associated with the object to determine the values, the expression is implemented in a particular programming language – (e.g., *debugger implementation in various programming languages and tools*

such as JAVA, HTML, Java Server Pages (JSP), Pascal, C#, C++, C, CGI, Perl, APIs, SDKs, assembly, firmware, microcode, and/or other languages and tools – see Dandoy, at least [0048] and [0049] -- with emphasis added).

As to claim 3, modified Dandoy with Kobayashi further discloses the programming language comprising at least one of C#, J# or Visual Basis.Net (e.g., *debugger implementation in various programming languages and tools such as JAVA, HTML, Java Server Pages (JSP), Pascal, C#, C++, C, CGI, Perl, APIs, SDKs, assembly, firmware, microcode, and/or other languages and tools – see at least [0048] and [0049] -- with emphasis added).*

As to claim 4, modified Dandoy with Kobayashi also discloses further comprising a plurality of expression evaluators, wherein each expression evaluator is associated with a different programming language (e.g., *the various component of debugger (e.g., debugger agent) can be implement in various programming languages and tools such as JAVA, HTML, Java Server Pages (JSP), Pascal, C#, C++, C, CGI, Perl, APIs, SDKs, assembly, firmware, microcode, and/or other languages and tools – see Dandoy at least [0048] and [0049] -- with emphasis added)..*

As to claim 5, modified Dandoy with Kobayashi also discloses the object comprises a class that includes at least one of a property or method (*e.g., Object properties may include, for example, class, position, size, visibilities, and other properties – see Dandoy at least [0041]).*

As to claim 6, modified Dandoy with Kobayashi further discloses the expression evaluator creates an instance of the display proxy associated (*e.g.*

proxy components 210, where each method in the component is converted into a separate proxy component of the components 210— see Kobayashi at least col. 8:10-14)

As to claim 8, modified Dandoy with Kobayashi the display proxy has access to private implementation specifies of the object. *(e.g., Parameters associated with a method are represented by properties of the proxy component created for that method. When each proxy component is displayed on the GUI of a conventional visual builder, its properties, and consequently, the parameters of the underlying method, are visually editable and can be bound visually to other component properties – See Kobayashi at least at least col. 5: 1-30, col. 7: 61-67, col. 8: 1-32, col. 4: 43-67).*

As to claim 9, modified Dandoy with Kobayashi discloses further comprising an attribute cache directory that stores an attribute associated with the display proxy, the expression evaluator employs the stored attribute to determine the details of the object *(e.g., the debug agent 115 is configured to collect execution data (i.e. object properties…) relating to the graphical user interface 105 during run-time, where the execution data can be obtained via reading state values stored in memory that are maintained by the application during execution – see Dandoy at least [0018] – with emphasis added).*

As to claim 10, modified Dandoy with Kobayashi further comprising an editing component that facilitates modifying a value associated with the object *(e.g., Parameters associated with a method are represented by properties of the proxy component created for that method. When each proxy component is*

displayed on the GUI of a conventional visual builder, its properties, and consequently, the parameters of the underlying method, are visually editable and can be bound visually to other component properties – See Kobayashi at least at least col. 5: 1-30, col. 7: 61-67, col. 8: 1-32, col. 4: 43-67).

As to claim 11, modified Dandoy with Kobayashi also discloses the variable display employs at least one attribute associated with the object that provides a format to display the determined values of one or more properties of the object *(e.g., the debug agent 115 can be configured to display/show properties of objects according to changing of certain colors or other properties (i.e., hierarchy (parent/child)) – see Dandoy at least [0025],[0026], and [0041-- with emphasis added*).

As to claim 12, modified Dandoy with Kobayashi discloses the attribute specifies at least one of whether a property of the object is displayed or how the property is displays *(e.g., the debug agent 115 can be configured to display/show properties of objects according to certain colors or other properties (i.e., hierarchy (parent/child)) – see Dandoy at least [0025], [0026], and [0041-- with emphasis added*).

As to claim 13, modified Dandoy with Kobayashi also discloses the attribute employs an enumeration to specify the format of the display (e.g., debug agent is configured to collect execution data relating to the graphical user interface, which includes object properties, events, and runtime states for *displaying/showing properties of objects according to certain colors or other*

properties (i.e., hierarchy (parent/child)) – see Dandoy at least [0018],[0025], [0026], and [0041]-- with emphasis added).

As to claim 14, modified Dandoy with Kobayashi further discloses one enumeration value that indicates the property should not be displayed to the developer *(e.g., the debug agent can determine the current state of the selected window and it changes its properties: hidden – see Dandoy at least [0025])*.

As to claim 15, modified Dandoy with Kobayashi further discloses the enumeration includes one enumeration value that indicates a hierarchical property is expanded by default –(*e.g., at any points during debugging, a hierarchy objects within the interface can be determined and displayed, the hierarchy can be displayed automatically see Dandoy, at least, [0046])*.

As to claim 16, modified Dandoy with Kobayashi also discloses the enumeration includes one enumeration value that indicates a hierarchical property is not expanded by default (*e.g., debugging requests may include a request to monitor events associated with an object … or request to hide or show and object see Dandoy, at least [0024])*.

As to claim 17, modified Dandoy with Kobayashi further discloses the enumeration includes one enumeration value that indicates a hierarchical property itself is not displayed and members of the hierarchical property are display (*e.g., at any points during debugging, a hierarchy objects within the interface can be determined and displayed – see Dandoy , at least, [0046])*.

As to claim 18, modified Dandoy with Kobayashi also discloses the attribute specifies what is displayed for a class *(e.g., the debug agent 115 can be*

configured to display/show properties of objects according to changing of certain colors or other properties (i.e., hierarchy (parent/child) within user request) – see Dandoy at least [0025],[0026], and [0041]– with emphasis added).

As per claim 21 and 26, Dandoy discloses a method facilitating attributed debugging comprising:

receiving a request to examine details of one or more properties of an object in a computer software application being debugged*(e.g.., executable application 100 (fig.1))* –*(e.g., object identity such name/type or hierarchy of objects can be determined and displayed upon user's requests – see at least [0046], [0017], [0018], and [0029] with emphasis added).*

It is noted that Dandoy does not explicitly disclose determining whether a display proxy attribute exist for the object, the display proxy is implemented as a private nested class of the object such that the display proxy within the definition of the object, the display proxy provides relevant properties regarding a state of the object and conceals properties related to implementation of the object; creating a display proxy for the object in accordance with display proxy attribute; and examining the display proxy in place of the object to determine debug information related to the object. However, Kobayashi, in analogous art, teaches

"In accordance with one illustrative embodiment, a proxy component (object proxy)is created for each method, including constructors (nested classes), in the component class code, which proxy component encapsulates the parameters of that method. In particular, parameters associated with a method are represented by properties of the proxy component created for that method. When each proxy component is displayed on the GUI of a conventional visual builder, its properties, and consequently, the parameters of the underlying method, are visually editable and can be bound visually to other component properties using, for example, a mechanism in a conventional visual

builder which links objects. Exceptions which occur during operation of the method are treated as events and can be visually passed to other components. Therefore, a program can be visually constructed and parameterized for runtime operation." – *See Kobayashi at least col. 5: 1-30, col. 7: 61-67, col. 8: 1-32, col. 4: 43-67 with emphasis added.*

Thus, it would have been obvious to one ordinary skill in the art to use visually creating and testing object of Kobayashi in Debug Agent 115 of Dandoy by creating proxy object and edit/test/ modifying object properties in user interface 105 for increasing programmer efficiency and reduce the time it takes to develop code as taught in Kobayashi (e.g. at least col. 4: 43-60).

Further regarding to claim 26, Dandoy discloses an attributes debugging system (*e.g.*, system of Fig. 1 – see at least [0017])) comprising: means for implementing the method regarding to claim 21 above.

As to claim 22, modified Dandoy with Kobayashi discloses further comprising providing the debug information determined by the display proxy to the developer in response to the request to examine the object –(*e.g., object identity such name/type or hierarchy of objects can be determined and displayed upon user's requests – see Dandoy, at least [0046], [0017], [0018], and [0029] with emphasis added).*

As to claim 23, modified Dandoy with Kobayashi discloses a computer readable medium (-- See Dandoy at least [0049]) having stored thereon computer executable instructions for carrying out the method of claim 21.

As to claim 27, Dandoy discloses a computer-implemented attributed debugging system (e.g., system of Fig. 1 – see at least [0017]) comprising:

a debugger (e.g., UI Debugger such as 110 or 200 of Figs 1 and 2) that facilitates debugging of a computer software application *(e.g.., executable application 100 (fig.1)) – See at least [0017], [0018], and [0029];*

an expression evaluator associated with the debugger that examines at least one object of the computer software application, the expression evaluator determines debug information comprising states of the at least one object, the states include values for at least one property of the at least one object--*(e.g., the debug agent 115 is configured to collect execution data (i.e. object properties…) relating to the graphical user interface 105 during run-time, where the execution data can be obtained via reading state values stored in memory that are maintained by the application during execution – See at least [0017], [0018], and [0029] with emphasis added*).

an attribute cache directory that retains instances of one more display proxies, the expression evaluator queries the attribute caches directory for instance of display proxies associated with the at least one object, the expression evaluator creates an instance if not retained in the attribute cache directory – *(e.g., the debug agent 115 is configured to collect execution data (i.e. object properties…) relating to the graphical user interface 105 during run-time, where the execution data can be obtained via reading state values stored in memory that are maintained by the application during execution (cache)– see at least [0018] – with emphasis added);*

a variable display component that presents the debug information to a developer, the debug information includes values of relevant properties of the at

least one object determined from examination of display proxies of the at least one object–(*e.g., object identity such name/type or hierarchy of objects can be determined and displayed upon user's requests – see at least [0046]*).

It is noted that Dandoy does not explicitly disclose the expression evaluator inspects the at least one object to verify if each of the at least one object includes a display proxy defined as private nested class of the object, the display proxy provides relevant properties of an associated object and conceals properties related to implementation of the object, the expression evaluator examines the display proxy in place of the object to determine debug information that includes value for the relevant properties of the at least one object; however, Kobayashi, in analogous art, teaches

"In accordance with one illustrative embodiment, a proxy component (object proxy) is created for each method, including constructors (nested classes), in the component class code, which proxy component encapsulates the parameters of that method. In particular, parameters associated with a method are represented by properties of the proxy component created for that method. When each proxy component is displayed on the GUI of a conventional visual builder, its properties, and consequently, the parameters of the underlying method, are visually editable and can be bound visually to other component properties using, for example, a mechanism in a conventional visual builder which links objects. Exceptions which occur during operation of the method are treated as events and can be visually passed to other components. Therefore, a program can be visually constructed and parameterized for runtime operation." – *See Kobayashi at least col. 5: 1-30, col. 7: 61-67, col. 8: 1-32, col. 4: 43-67 with emphasis added.*

Thus, it would have been obvious to one ordinary skill in the art to use visually creating and testing object of Kobayashi in Debug Agent 115 of Dandoy by creating proxy object and edit/test/ modifying object properties in user

interface 105 for increasing programmer efficiency and reduce the time it takes to develop code as taught in Kobayashi (e.g. at least col. 4: 43-60).

6. Claims 19 and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Dandoy, (U.S. Patent No. 2004/0230954 A1) in view of Kobayashi (US 6, 633,888 B1) , and in further view of Bates et al., (U.S. Patent No. 6,961,924 B2 of record – hereinafter Bates)

As to claim 19, it is noted that modified Dandoy with Kobayashi does not explicitly disclose the attributes includes an argument that comprises a string that is displayed in a value column for an instance of the class. However, Bates, in an analogous art teaches, the debugger 123 determines whether any attributes are set for the variable. If any attributes you set for the variable, then processing to step 616 where the appropriate attribute indicator (e.g., G, S, I, R, C, P) for each set attribute is associated with the variable value and determines whether the variable value is associated with a fields of a record (class) in a symbol table 120, that is the debugger determines whether a field is to be displayed*(see step 614 and 618, Fig. 6, col. 11: 32-50; also see attributes are set in the fields 312-322, Fig. 3 and associated text, and step 616, col. 11: 34-37 – with emphasis added).*

It would have been obvious to one ordinary skill in the art at the time invention was made to use the attribute indicator of the debugger 123 of Bate in debug agent of modified Dandoy and Kobayashi as convenient way of accessing run time variable value to user as taught Bates *(e.g. step 616, col. 11: 34-37).*

As to claim 20, modified Dandoy with Kobayashi and Bates discloses the

argument is associated with a property of the class *(e.g., at step 618, the debugger 123 determines whether the variable value is associated with a field of a record (class) –see Bates , at least col. 11: 41-42).*

Conclusion

8. The prior art made of record and not relied upon is considered pertinent to the applicant disclosure.

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Marina Lee whose telephone number is (571) 270-1648. The examiner can normally be reached on M-F (11am-7: 30pm) Est.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service

Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/M. L./
Examiner, Art Unit 2192

/Tuan Q. Dam/
Supervisory Patent Examiner, Art Unit 2192